IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Serial No.:	10/670,898	Conf. No.:	5987
Filing Date:	09/25/2003	Art Unit:	2193
Applicant:	Masek et al.	Examiner:	Mitchell, Jason D.
Title:	Method, system and program product for testing a server using a reentrant test application	Docket No.:	LOT920030024US1 (IBML-0027)

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

BRIEF OF APPELLANTS

This is an appeal from the Final Rejection dated December 10, 2010, rejecting claims 1-9, 11-18, and 20-26. This Brief is accompanied by the requisite fee set forth in 37 C.F.R. 1.17(c).

REAL PARTY IN INTEREST

International Business Machines Corp. is the real party in interest.

RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences.

STATUS OF CLAIMS

As filed, this case included claims 1-26. Claims 1-9, 11-18 and 20-26 remain pending, and claims 10 and 19 have been previously cancelled. Claims 1-9, 11-18, and 20-26 stand rejected and form the basis of this appeal.

STATUS OF AMENDMENTS

No amendment to the claims has been submitted subsequent to the December 10, 2010 Final Office Action.

SUMMARY OF CLAIMED SUBJECT MATTER

Claim 1 claims a method for testing a server application using a reentrant test application (e.g., Para. 16), comprising: providing (e.g., S1, Figure 4; Para. 26) a test application (e.g., 43, Figure 1; Para. 18) that satisfies reentrancy requirements (e.g., Para. 17) on a client (e.g., 12, Figure 1; Para. 18); identifying (e.g., S2, Figure 4; Para. 26) application protocol interfaces (APIs) (e.g., Para. 22) associated with the test application (e.g., 43, Figure 1); providing (e.g., S3, Figure 4; Para. 26) a test script capable of invoking the APIs (e.g., Para. 22); and instantiating, via the test script, (e.g., S4, Figure 4; Para. 26) a plurality of instances of the test application (e.g., 43, Figure 1) using threads (e.g., Para. 23), wherein the instantiating and execution of each of the plurality of instances of the test application occur within a single process (e.g., 32, Figure 1; Para. 23), sharing all services and memory space which are exclusively dedicated to the single process (e.g., 32, Figure 1) with others of the plurality of instances (e.g., Para. 17 and 23), without requiring multiple processes to instantiate the plurality of instances within (e.g., Para. 24).

Claim 9 claims a system for testing a server application (e.g., Para. 16), comprising an interface identification system (e.g., 38, Figure 1; Para. 22) for identifying (e.g., S2, Figure 4; Para. 26) application protocol interfaces (APIs) (e.g., Para. 22) associated with a test application (e.g., 43, Figure 1); and an application instantiation system (e.g., 40, Figure 1; Para. 212) for instantiating (e.g., S4, Figure 4; Para. 26) a plurality of instances of the test application (e.g., 43, Figure 1) on a client (e.g., 12, Figure 1; Para. 18) using threads (e.g., Para. 23) if the test application satisfies reentrancy requirements (e.g., Para. 17), wherein the test application instantiation system (e.g., 40, Figure 1; Para. 212) comprises a driver that executes a test script capable of invoking the identified APIs (e.g., Para. 23), and wherein the instantiating and execution of each of the plurality of instances of the test application (e.g., 43, Figure 1), via the test script, occur within a single process (e.g., 32, Figure 1; Para. 23), sharing all services and memory space which are exclusively dedicated to the single process (e.g., 32, Figure 1) with others of the plurality of instances (e.g., Para. 17 and 23), without requiring multiple processes to instantiate the plurality of instances within (e.g., Para. 24).

Claim 18 claims a program product stored on a recordable medium (e.g., Para. 27) for testing a server application (e.g., Para. 16), which when executed, comprises program code for identifying (e.g., S2, Figure 4; Para. 26) application protocol interfaces (APIs) (e.g., Para. 22) associated with the test application (e.g., 43, Figure 1); program code for instantiating (e.g., S4, Figure 4; Para. 26) a plurality of instances of a test application (e.g., 43, Figure 1) on a client (e.g., 12, Figure 1; Para. 18) using threads (e.g., Para. 23) if the test application satisfies reentrancy requirements (e.g., Para. 17), wherein the program code for instantiating executes a test script capable of invoking the identified APIs (e.g., Para. 23), and wherein the instantiating and execution of each of the plurality of instances of the test application (e.g., 43, Figure 1), via

the test script, occur within a single process (e.g., 32, Figure 1; Para. 23), sharing all services and memory space which are exclusively dedicated to the single process (e.g., 32, Figure 1) with others of the plurality of instances (e.g., Para. 17 and 23), without requiring multiple processes to instantiate the plurality of instances within (e.g., Para. 24).

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

I. Claims 1-9, 11-18 and 20-26 are rejected under 35 U.S.C. §112, first paragraph as allegedly failing to comply with the written description requirement.

II. Claims 1-5, 7-9, 11-14, 16-18, 20-23 and 25-26 are rejected under 35 U.S.C. §103(a) as allegedly being unpatentable over Duggan et al. (U.S. Patent No. 6,002,871), hereafter "Duggan," in view of Dinker et al. (U.S. Patent Pub. No. 20040199815), hereafter "Dinker," in view of Partamian et al. (U.S. Patent No. 7,062,755), hereafter "Partamian," in view of Firth et al. (U.S. Patent No. 5,987,517), hereafter "Firth."

III. Claims 6, 15, and 24 are rejected under 35 U.S.C. §103(a) as allegedly being unpatentable over Duggan in view of Dinker in view of Partamian in view of Firth in further view of Lindholm et al. ("The Java™ Virtual Machine Specification"), hereafter "Lindholm."

ARGUMENT

I. Rejection of claims 1-9, 11-18 and 20-26 under 35 U.S.C. §112, first paragraph

Appellant respectfully submits that the rejection under 35 U.S.C. 112, second paragraph, is defective and requests reversal thereof for the following reasons. Appellants respectfully submit that it is not necessary that the exact wording from the claims be found in the specification as the Examiner appears to require, but rather "…newly added claim limitations must be supported in the specification through express, implicit, or inherent disclosure." MPEP 2163(I)(B).

In the Final Office Action, the Examiner contends that claim 1 is allegedly not described in the specification. (Final Office Action, Page 6). The Examiner cites Appellants specification at Para. 17, wherein the specification states that "multiple test applications will be running in the same process, sharing the same services and memory space." (Final Office Action, Page 6). Further, the Examiner asserts that there is no disclosure of exclusively dedicated services or memory. However, Appellants assert that this feature would be clear to one skilled in the art. For instance, the specification describes that "the test application on a client that is driving load against the server application should satisfy reentrancy requirements. Specifically, as will be further described below, the present invention will instantiate multiple instances of the test application to test the server application. As such, multiple test applications will be running in the same process, sharing the same services and memory space." (Para. 17). Clearly, Appellants' disclosure states that in order to test such a load, the tests will be running in the same process and sharing the same services and memory space. Since the testing is specifically based on the load generated, it would be clear to one of ordinary skill in the art that the services and memory are dedicated to this testing, or the results would not be conclusive. To this extent,

Appellants assert that these portions of the specification, *inter alia*, implicitly and/or inherently support the claimed limitation objected to by the Examiner. Accordingly, Appellants respectfully request that the Examiner's rejection be overruled.

In light of the above, Appellants respectfully request reversal of the Examiner's rejection of claim 1, as well as claims 2-8 which were rejected for their dependency. Further, Appellants request reversal of the rejection of claims 9, 11-18, and 20-26 which were rejected based on the rejection of claim 1.

II. Rejection of claims 1-5, 7-9, 11-14, 16-18, 20-23 and 25-26 under 35 U.S.C. §103(a) as allegedly being unpatentable over Duggan in view of Dinker in view of Partamian in view of Firth

The rejection under 35 U.S.C. 103(a) is defective because the cited combination fails to teach or suggest each and every feature of the claims.

Independent Claim 1

Regarding independent claim 1, Appellants submit that the Examiner fails to show that Duggan in view of Dinker, Partamian, and Firth teaches or suggests, *inter alia*, "instantiating, via the test script, a plurality of instances of the test application using threads, wherein the instantiating and execution of each of the plurality of instances of the test application occur within a single process." The Examiner admits that Duggan fails to teach or suggest that the plurality of instances occur within a single process, and instead asserts that this feature is taught by Dinker. The Examiner posits that Dinker teaches each test agent 110 may be implemented as a multithreaded application, citing Par. 0028 and 0031 of Dinker, and that using the single agent

of Duggan with the multithreaded agents of Dinker teaches the features of the claim. (Final Office Action, Pages 8-9). However, Appellants respectfully disagree with the combination of the references. For instance, as is clearly stated in Dinker, the test agents 110 are consistently a plurality of test agents, not a single agent. Dinker in fact teaches away from the single test agent of Duggan, stating that "by implementing test cluster 100 from several multi-threaded test processes (i.e. test agents 110)" there is "less thread starvation than if the same number of clients were simulated by a single multi-threaded process." (Dinker, Para. 0031). As such, Dinker viewed alone or in combination with Duggan fails to teach or suggest the feature of instantiating each of the plurality of instances of the test application within one process, and rather teaches directly away from this feature, specifying that a single multi-threaded process would result in thread starvation as cited above. Therefore, there would have been no motivation to combine the teachings of Duggan and Dinker, as Dinker specifically suggests against a single process. Appellants assert that the Examiner is using Appellants' own disclosure as motivation to combine the references.

With further respect to independent claim 1, Appellants respectfully assert, in addition to the above arguments, that the cited references also fail to teach or suggest, *inter alia*, "sharing all services and memory space exclusively dedicated to the single process with others of the plurality of instances." Claim 1. The Examiner admits that Duggan and Dinker do not explicitly disclose sharing all services and memory space among the plurality of instances. Rather, the Examiner cites a passage of Partamian that teaches a general description of processes and threads, specifically citing that "threads in the same process share information **using** memory, atomic structures, mutexes, semaphores, etc." (Partamian, Col. 3, lines 12-17, emphasis added). Applicants, however, assert that the Examiner has misinterpreted the passage cited, as

emphasized above. The passage states that threads within a process share information by using memory, not that they share a memory space, or even that there is a shared memory at all. There is further no reference to sharing services. Appellants further assert that there is no disclosure that the services and memory space are dedicated exclusively for the single process. As such, Partamian clearly fails to teach or suggest this feature, as sharing information using memory clearly is not equivalent to services and memory space dedicated for a single process. As such, the cited combination fails to teach or suggest that all services and memory space are shared among the plurality of instances, and Fink fails to cure this deficiency.

With further respect to independent claim 1, Appellants continue to respectfully assert, in addition to the above arguments, that the cited references also fail to teach or suggest, *inter alia*, "… identifying application protocol interfaces (APIs) …, prior to the instantiating step…[and] providing a test script capable of invoking the APIs …". Claim 1. The Office admits that Duggan, Dinker, and Partamian do not explicitly disclose that its command module is implemented as APIs. Rather, the Office continues to cite a passage of Firth that teaches, generically, that APIs exist, reciting "functions in the Internet API reside in a dynamic link library (DLL)." (Final Office Action, Page 10-11 citing Firth, Col. 2, lines 63-67). As a first example, this reference describing API simply describes what an API is with no reference to test applications or any related information. Further, as Appellants have repeatedly argued, the Examiner attempts to replace whole pages of Duggan that describe the formation of scripts with one generic sentence describing where an API is stored. Appellants submit that this combination is faulty as Duggan's entire disclosure relies upon the use of Visual Basic 5.0. For instance, Col. 14 of Duggan using Visual Basic is not the only reference to the program, rather the entire description of code up to Col. 21 is a description of the code used with Visual Basic. This

description is preceded by Col. 13 of Duggan describing that the Test Tool Program also uses Visual Basic, and that the code modules and forms are the building blocks of the program. (Duggan, Col. 13, Lines 22-41). As such, entire pages of Duggan describe the use of Visual Basic, which is the Microsoft Program used in the entirety of the application. In fact, at Page 11 of the Final Office Action, the Examiner quotes the ease of use of Duggan with no knowledge of underlying programmed instruction of the command module needed, citing Col. 14, Lines 2-4 of Duggan. Appellants note that this is in reference to the preferred embodiment described in both Col. 13 and Col. 14, which utilizes the Visual Basic program for such an ease of use. Applicants respectfully submit that the reference to an API in Firth has nothing to do with creating testing of application programs, as in Duggan, and any attempt to incorporate this generic API into the Visual Basic GUI based system of Duggan would in fact, at best, lead to undue experimentation and yield unpredictable results. In fact, the Examiner argues that a generic teaching that APIs exist is sufficient to show that implementing Duggan's test environment in APIs was obvious, quoting which file formats to use. (Final Office Action, Page 5). Appellants assert that the Examiner is using personal knowledge rather than relying upon prior art, and the rejection is thus faulty. Accordingly, Appellants request that the rejection of claim 1 be reversed.

As such, Appellants assert that Duggan in view of Dinker, Partamian, and Firth, viewed alone or in combination, do not teach or suggest each of the features discussed above. Accordingly, the cited combination fails to teach or suggest each and every feature of the claimed invention.

Since the Examiner fails to show that Duggan in view of Dinker, Partamian, and Firth teaches or suggests each and every feature of claim 1, Appellants respectfully submit that the rejection is deficient.

In light of each of the above reasons, Appellants submit that the Examiner fails to establish a *prima facie* case of obviousness with regard to claim 1 and Appellants respectfully request reversal of the rejection over Duggan in view of Dinker, Partamian, and Firth.

Claims 2-5 and 7-8

Since claims 2-5 and 7-8 depend from claim 1, which Appellants have argued *supra* to not be obvious over Duggan in view of Dinker, Partamian, and Firth under 35 U.S.C. § 103(a), Appellants maintain that claims 2-5 and 7-8 are likewise not obvious over Duggan in view of Dinker, Partamian, and Firth under 35 U.S.C. § 103(a) for at least the reasons given above, as well as their own unique features.

Independent Claim 9

With respect to independent claim 9, Appellant submits that the Examiner fails to show that Duggan in view of Dinker, Partamian, and Firth teaches or suggests each and every feature of the claimed invention.

For example, for reasons that should be clear from the discussion of Duggan in view of Dinker, Partamian, and Firth above, Appellant submits that the cited reference fails to disclose the system of claim 9. For at least the reasons discussed above, the cited combination fails to teach or suggest each and every feature of independent claim 9. Appellants further assert that the claim is allowable for its own unique features.

Accordingly, Appellant respectfully requests reversal of the rejection of claim 9 as allegedly being unpatentable over Duggan in view of Dinker, Partamian, and Firth.

Claims 11-14 and 16-17

Since claims 11-14 and 16-17 depend from claim 9, which Appellants have argued *supra* to not be obvious over Duggan in view of Dinker, Partamian, and Firth under 35 U.S.C. § 103(a), Appellants maintain that claims 11-14 and 16-17 are likewise not obvious over Duggan in view of Dinker, Partamian, and Firth under 35 U.S.C. § 103(a) for at least the reasons given above, as well as their own unique features.

Independent Claim 18

With respect to independent claim 18, Appellant submits that the Examiner fails to show that Duggan in view of Dinker, Partamian, and Firth teaches or suggests each and every feature of the claimed invention.

For example, for reasons that should be clear from the discussion of Duggan in view of Dinker, Partamian, and Firth above, Appellant submits that the cited reference fails to disclose the program product of claim 18. For at least the reasons discussed above, the cited combination fails to teach or suggest each and every feature of independent claim 18. Appellants further assert that the claim is allowable for its own unique features.

Accordingly, Appellant respectfully requests reversal of the rejection of claim 18 as allegedly being unpatentable over Duggan in view of Dinker, Partamian, and Firth.

Claims 20-23 and 25-26

Since claims 20-23 and 25-26 depend from claim 18, which Appellants have argued *supra* to not be obvious over Duggan in view of Dinker, Partamian, and Firth under 35 U.S.C. § 103(a), Appellants maintain that claims 20-23 and 25-26 are likewise not obvious over Duggan in view of Dinker, Partamian, and Firth under 35 U.S.C. § 103(a) for at least the reasons given above, as well as their own unique features.

III. Rejection of claims 6, 15, and 24 as allegedly being unpatentable Duggan in view of Dinker, Partamian, and Firth in further view of Lindholm

The rejection under 35 U.S.C. 103(a) is defective because the references fail to teach or suggest each and every feature of the claims.

Claims 6, 15, and 24

Since claims 6, 15, and 24 depend from claims 1, 9, and 18, which Appellants have argued *supra* to not be obvious over in view of Duggan in view of Dinker, Partamian, and Firth under 35 U.S.C. § 103(a), Appellants maintain that claims 6, 15, and 24 are likewise not obvious over Duggan, Dinker, Partamian, and Firth in further view of Lindholm under 35 U.S.C. § 103(a).

Further, Appellants assert that Lindholm does not cure the deficiencies of the cited combination. For instance, while Lindholm discloses a generic Java™ Virtual Machine, it does not disclose the instantiation of multiple instantiated threads within a single process for use in testing a server application. Appellants respectfully submit that a person of ordinary skill applying Lindholm in combination with Duggan will not be able to achieve the results of the claimed invention without undue experimentation, at best. The Examiner's proposed combination of Duggan and Lindholm would lead to unpredictable results. As such, the combined teachings of Duggan, Dinker, Partamian, Firth and Lindholm do not support the Examiner's assertion of a prima facie obviousness. Accordingly, Appellants respectfully request that the Office withdraw its rejection.

V. Conclusion

In summary, Appellants submit that independent claims 1, 9, and 18 are allowable over the cited art because the Examiner's use of Duggan, Dinker, Partamian, and Firth fails to present a *prima facie* showing that each element of the claimed inventions is taught or suggested by the cited art. Additionally, Appellants respectfully submit that all other pending claims are allowable over the cited art by, *inter alia*, dependency.

Respectfully submitted,

/Nathan B. Davis/

Nathan B. Davis, Reg. No. 67,474
Hoffman Warnick LLC
75 State Street, 14th Floor
Albany, NY 12207
(518) 449-0044 - Telephone
(518) 449-0047 - Facsimile

Dated: 09 May 2011

CLAIMS APPENDIX

Claim Listing:

1. A method for testing a server application using a reentrant test application, comprising:

providing a test application that satisfies reentrancy requirements on a client;

identifying application protocol interfaces (APIs) associated with the test application;

providing a test script capable of invoking the APIs; and

instantiating, via the test script, a plurality of instances of the test application using threads, wherein the instantiating and execution of each of the plurality of instances of the test application occur within a single process, sharing all services and memory space which are exclusively dedicated to the single process with others of the plurality of instances, without requiring multiple processes to instantiate the plurality of instances within.

2. The method of claim 1, wherein upon execution, the test script instantiates the plurality of instances of the test application using threads within the single process.

3. The method of claim 1, wherein the server application is a network application.

4. The method of claim 1, wherein the reentrancy requirements dictates that the plurality of instances of the test application be run within the single process without interfering with each other.

5. The method of claim 1, wherein each of the plurality of instances of the test application corresponds to a separate thread, and wherein each of the separate threads is associated with a different connection to the server.

6. The method of claim 1, wherein the process comprises a JAVA™ virtual machine.

7. The method of claim 1, wherein the plurality of instances of the test application simulate use of the server application by a plurality of users.

8. The method of claim 1, further comprising collecting data corresponding to the test.

9. A system for testing a server application, comprising

an interface identification system for identifying application protocol interfaces (APIs) associated with a test application; and

an application instantiation system for instantiating a plurality of instances of the test application on a client using threads if the test application satisfies reentrancy requirements, wherein the test application instantiation system comprises a driver that executes a test script capable of invoking the identified APIs, and wherein the instantiating and execution of each of the plurality of instances of the test application, via the test script, occur within a single process, sharing all services and memory space which are exclusively dedicated to the single process with others of the plurality of instances, without requiring multiple processes to instantiate the plurality of instances within.

11. The system of claim 9, wherein upon execution, the test script instantiates the plurality of instances of the test application using threads within the single process.

12. The system of claim 9, wherein the reentrancy requirements dictates that the plurality of instances of the test application be run within the single process without interfering with each other.

13. The system of claim 9, wherein each of the plurality of instances of the test application corresponds to a separate thread, and wherein each of the separate threads is associated with a different connection to the server.

14. The system of claim 9, wherein the application is a network application.

15. The system of claim 9, wherein the process comprises a JAVA™ virtual machine.

16. The system of claim 9, wherein the plurality of instances of the test application simulate use of the server application by a plurality of users.

17. The system of claim 9, further comprising a data collection system for collecting data corresponding to the test.

18. A program product stored on a recordable medium for testing a server application, which when executed, comprises

program code for identifying application protocol interfaces (APIs) associated with the test application;

program code for instantiating a plurality of instances of a test application on a client using threads if the test application satisfies reentrancy requirements, wherein the program code for instantiating executes a test script capable of invoking the identified APIs, and wherein the instantiating and execution of each of the plurality of instances of the test application, via the test script, occur within a single process, sharing all services and memory space which are exclusively dedicated to the single process with others of the plurality of instances, without requiring multiple processes to instantiate the plurality of instances within.

20. The program product of claim 18, wherein upon execution, the test script instantiates the plurality of instances of the test application using threads within the single process.

21. The program product of claim 18, wherein the reentrancy requirements dictates that the plurality of instances of the test application be run within the single process without interfering with each other.

22. The program product of claim 18, wherein each of the plurality of instances of the test application corresponds to a separate thread, and wherein each of the separates threads is associated with a different connection to the server.

23. The program product of claim 18, wherein the server application is a network application.

24. The program product of claim 18, wherein the process comprises a JAVA™ virtual machine.

25. The program product of claim 18, wherein the plurality of instances of the test application simulate use of the server application by a plurality of users.

26. The program product of claim 18, further comprising program code for collecting data corresponding to the test.

EVIDENCE APPENDIX

No evidence has been entered and relied upon in the appeal.

RELATED PROCEEDINGS APPENDIX

No decisions rendered by a court or the Board in any proceeding are identified in the related appeals and interferences section.